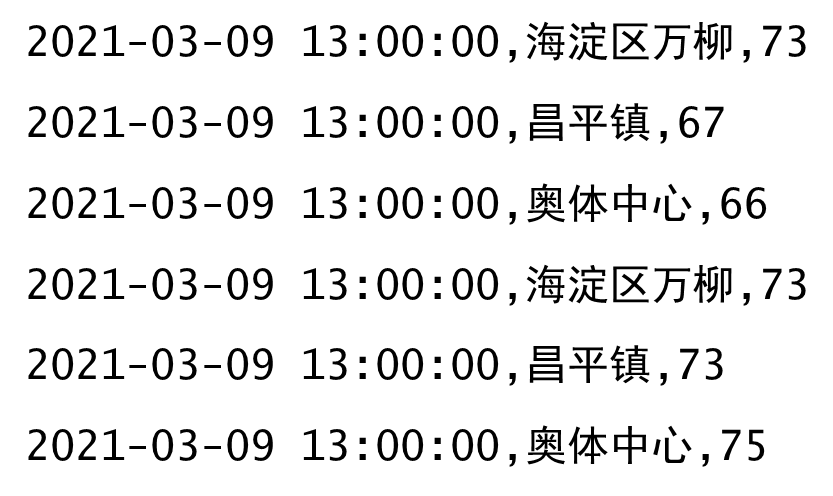
# Linux上机作业1实验报告

## 实验题目

从因特网上搜索Web页，用wget获取网页，处理网页html文本数据，从中提取出当前时间点北京各监测站的PM2.5浓度，输出如下CSV格式数据：



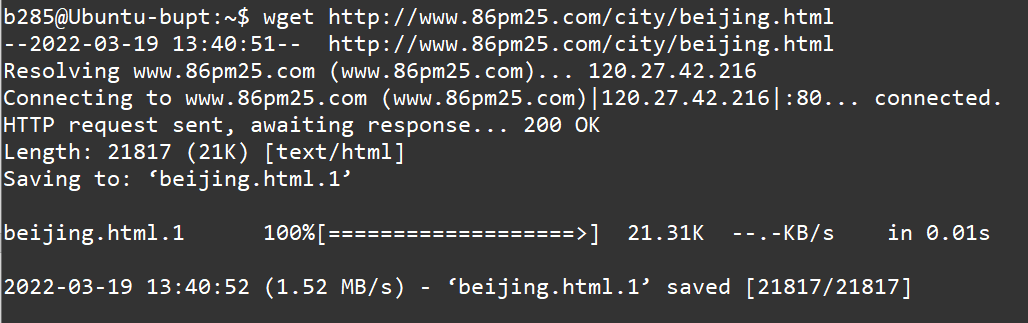
## 实验步骤

* 1. 在如下网址中发现有我们想要的数据：

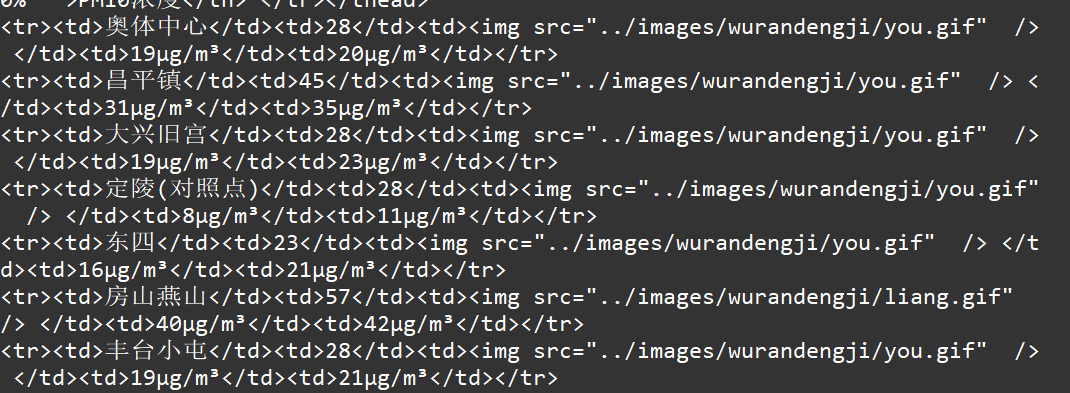
<http://www.86pm25.com/city/beijing.html>



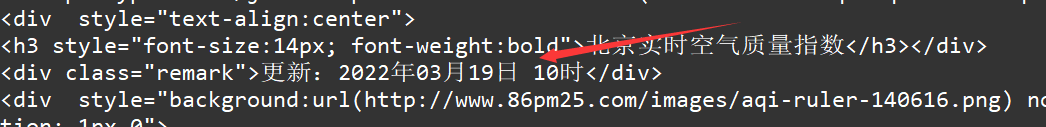
* 1. 打开putty，在命令行中使用wget命令获取这个页面：



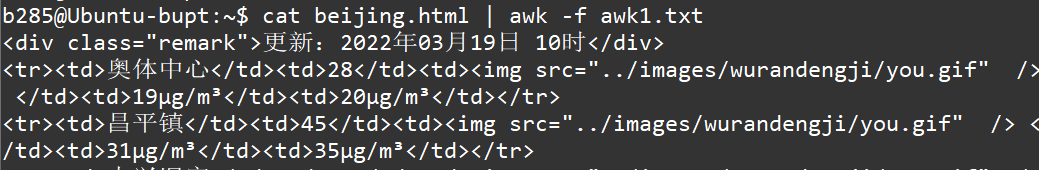
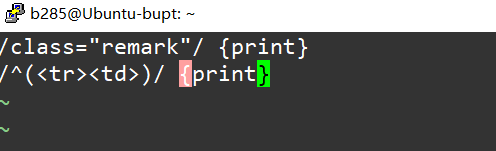
* 1. 由于之前已经获取了这个网页，所以这里将文件命名为“beijing.html.1”， 将这个文件删除，只保留原来的“beijing.html”文件，现在我们可以使用cat命令查看这个文件：



在这里我们看见了我们想要的监测站点的各个数据，并且以表格方式存储，但是时间不包含在其中，继续查找，我们可以找到时间的位置：

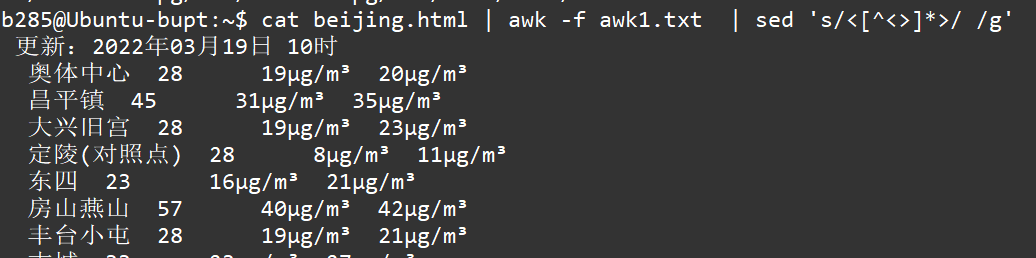


* 1. 可以看出，时间放在了div的盒子里，而根据这个div的类“class=“remark””，因此，再加上之前我们看到的每一个观测点的开头为<tr><td>，故可以创建一个awk1.txt文件，将时间行，与观测点行选择出来：



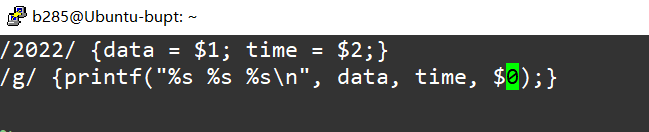
这样我们就看到我们需要的数据已经整合在一起了。

* 1. 接下来我们需要去掉所有的标签，因为标签都是使用一对尖括号括起来的，所以可以使用sed语句，将尖括号及其内部的内容整体替换为空格：

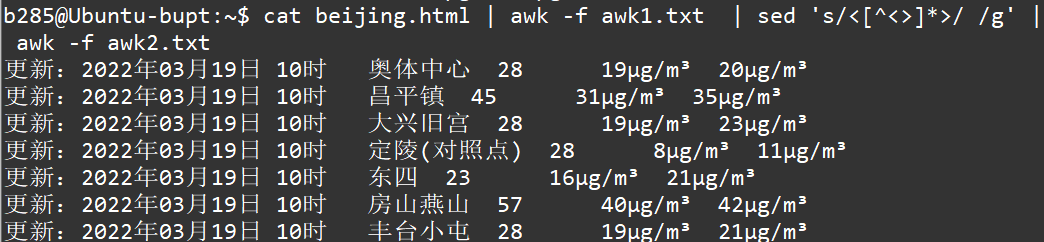


其中的正则表达式<[^<>]\*>的含义为以<开头，以>结尾，中间任意个不为<>的字符若干个组成的字符串，这样一来可以看到数据清晰了许多。

* 1. 下一步可以将时间信息与每一行进行合并，方便我们使用sed对每一行进行格式化操作，新建文件awk2.txt：

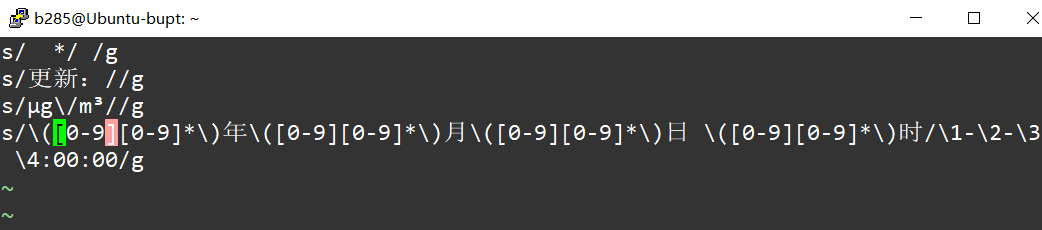


执行后得到的输出为：

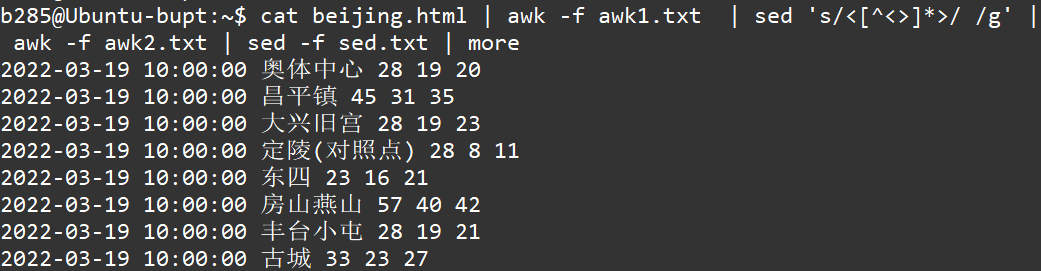


这样我们就将我们需要的数据放在了同一行中，便于接下来使用sed进行每一行字符串的格式化。

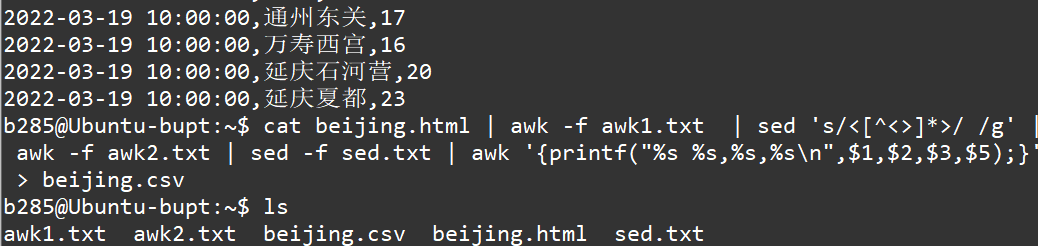
* 1. 首先需要将每一行最开始的“更新：”这三个字符和污染指数的单位“μg/m³”删除，然后将每一行中各个数据间的空格数量置为1，最后进行时间的格式化操作，我么可以新建一个sed.txt文件来存放刚才的这三个步骤：



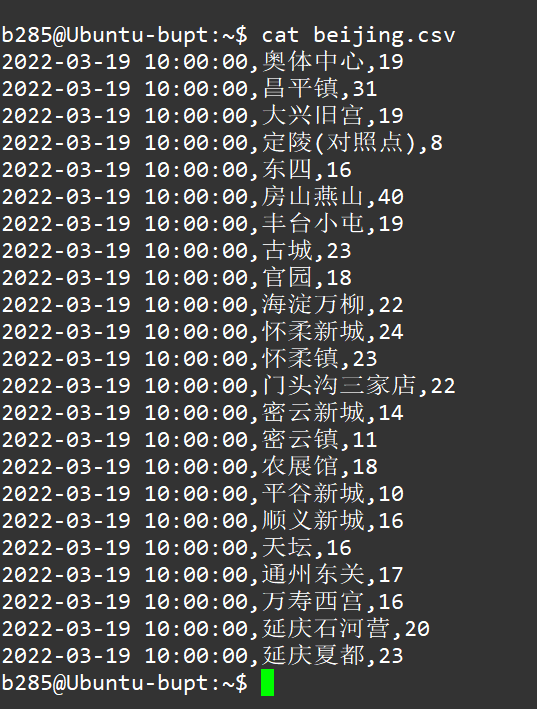
这样执行后得到了如下输出：



* 1. 再次使用awk命令选择我们想要的pm2.5的值（由原网站可知第五列数据为pm2.5的值），并在各个属性之间加上逗号，最后重定向到一个beijing.csv文件中：

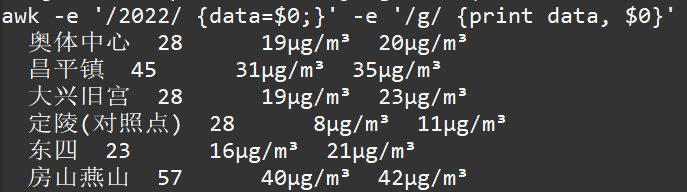


* 1. 最后我们查看这个文件内容：

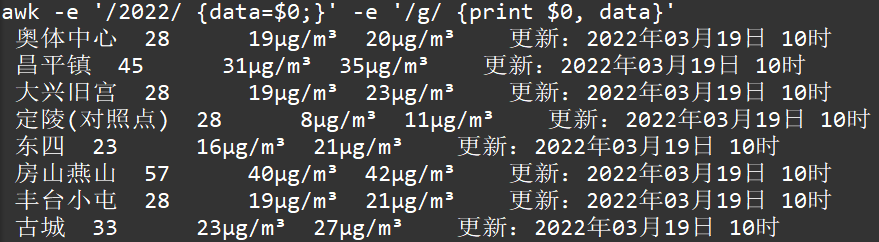


## 实验总结

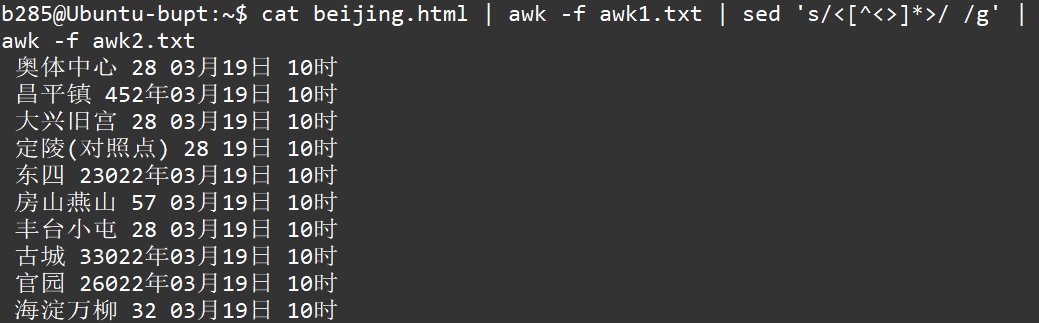
本次实验让我熟悉了Linux中正则表达式的操作以及sed、awk等命令的使用，最重要的训练了写正则表达式的能力以及处理一段文本时的逻辑顺序，在实验期间也走了不少弯路导致处理数据起来十分繁琐，在几次修改后已经得到简化，然而还有一些问题没有得到解决，比如在写awk2.txt文件的时候一开始我将日期设置为变量data，但是到了最后输出的时候却看不见data的值，如图：



然而将data与$0的顺序调换后发现又可以输出了：



最后测试发现第一次不是没有输出日期而是监测点的信息将时间信息覆盖掉了：



从结果来看更像是使用变量data输出$0的时候光标没有发生移动，导致数据之间覆盖，这个问题还没有解决。

通过本次实验，我认识到了正则表达式在处理字符串的强大之处，如果能更加熟练掌握这个技术想必在之后的学习中发现不少捷径。